Assembly, Integration, & Evolution Overview

Howard Lipson, Software Engineering Institute [vita[1]]

2005-09-28

The objective of the Assembly, Integration & Evolution content area is to raise awareness about the essential technical, business, and individual user issues that must be addressed during assembly, integration, and evolution to achieve and maintain a high degree of system-wide assurance of security and survivability.

Acknowledgement. Thanks to Gunnar Peterson of Cigital for his contribution to this overview.

The design, acquisition, and implementation decisions made during the assembly and integration of components and services into a larger system clearly have a profound impact on the security and survivability of the system as a whole. The objective of the Assembly, Integration & Evolution content area is to raise awareness about the essential technical, business, and individual user issues that must be addressed during assembly, integration, and evolution to achieve and maintain a high degree of system-wide assurance of security and survivability.

During assembly and integration, the logical design assumptions for a system meet the physical, business, technical, organizational, and individual user realities of the target system environment. Current trends point toward a sharp increase in exploits based on assembly-integration design errors, architectural mismatches among components, insecure identity management and services, false assumptions about a component's properties, an over-reliance on perimeter-based network security mechanisms, and the use of components in contexts (environments) not envisioned by the components' designers. Business pressures for increased efficiency and flexibility are moving applications toward "just-in-time" service creation and delivery (for example, through dynamic assembly in a web services environment), and are therefore stressing the limits of security and survivability even further. User privacy concerns centering on what identifiable information will be used for tracking and tracing may create constraints and conflict with security goals. The system-wide effects of the emergent behavior of large numbers of (software) components and services are mapped onto the production infrastructure. Unfortunately, how to compose the security and survivability properties of these components and services in a trustworthy manner is poorly understood by the software engineering and research communities. The problematic effects of emergent behavior are accentuated in the Service Oriented Architecture and Web Services paradigms, which rely on loose coupling and do not lend themselves to comprehensive end-to-end testing. One of the primary goals for this content area is to serve as a roadmap outlining the fundamental software assurance challenges and design considerations posed by assembly, integration, and evolution for managers, software engineers, and researchers.

This is the first release of material for this content area. New documents will be added over time and some of the existing documents will continue to evolve. This initial set of documents is not meant to be comprehensive or to give exhaustive coverage of this area. Rather, these documents are a sample of things to come. The initial set of documents for this content area consists of the following:

- Application Firewalls & Proxies – Introduction and Concept of Operations[2]: Providing a secure operating environment for business-critical applications is among the most crucial steps in the assembly and integration process. This document describes one of the many potential topic areas involving the integration of business applications into a supporting IT security infrastructure.

1. daisy:15 (Lipson, Howard F.)
2. daisy:30 (Application Firewalls and Proxies - Introduction and Concept of Operations)

Application firewalls attempt to use application-specific knowledge to improve the perimeter defense that the security infrastructure provides.

- Assembly and Integration Case Study: Enterprise Patch Management[3]: Successfully managing the inevitable changes to an enterprise-wide application is a key aspect of assembly and integration. “If these changes aren't properly managed across platforms and throughout each of the stages of the software development life cycle, production failures, including security problems, can result.” This document presents a case study of a Fortune 500 company where deficiencies in patch management left many of the company’s servers vulnerable to cyber attack and subsequent infection by the Slammer worm.
- Evolutionary Design of Secure Systems – The First Step is Recognizing the Need for Change[4]: A fundamental truth of system design is that, in the absence of countermeasures, a system's security will degrade over time. Changes in the environment or usage of a system, or changes to the elements that compose the system, often introduce new or elevated threats that the system was not designed to handle and is ill-prepared to defend itself against. The first step in evolving to meet new threats to your system's security is to recognize the need for change-that is, the need to enter the evolution phase of the system development life cycle.
- Identity in Assembly and Integration[5]: Securely integrating a shared service across highly distributed software systems presents a significant challenge at every phase of the software development life cycle. Moreover, there is a crucial need within the project team(s) for common abstractions and a common understanding of all the relevant aspects of a shared service. This document discusses the issues and necessary abstractions related to integrating identity services, which are particularly critical as the basis for granting or denying access to system resources and data.
- Trustworthy Composition: The System is Not Always the Sum of Its Parts[6]: This document surveys several of the profound technical problems and challenges faced by practitioners in the assembly and integration of secure and survivable systems. “It is critical that the practitioner understands the limitations of current techniques and hence maintains a healthy skepticism about the assurance associated with a complex software-intensive system, as well as for any ‘silver bullets’ proposed to mitigate that complexity.”

SEI Copyright

3. daisy:206 (Assembly and Integration Case Study: Enterprise Patch Management)
4. daisy:467 (Evolutionary Design of Secure Systems - The First Step Is Recognizing the Need for Change)
5. daisy:207 (Identity in Assembly and Integration)
6. daisy:50 (Trustworthy Composition: The System is Not Always the Sum of Its Parts)

1. http://www.sei.cmu.edu/about/legal-permissions.html

Naam	Waarde
Copyright Holder	SEI

Velden

Naam	Waarde
is-content-area-overview	true
Content Areas	Best Practices/Assembly, Integration, & Evolution
SDLC Relevance	Architecture Design
Workflow State	Publishable